

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

学 校 复旦大学

参赛队号 20102460127

队员姓名	1. 徐瑾
	2. 肖晗微
	3. 宋凤丽

中国研究生创新实践系列大赛

“华为杯”第十七届中国研究生

数学建模竞赛

题 目 ASIC 芯片上的载波恢复 DSP 算法设计与实现

摘 要：

光数字信号处理（DSP）芯片通常基于专用集成电路（ASIC）实现，DSP 算法设计通常包含两个步骤，第一步根据信道损伤的物理模型设计补偿算法，第二步根据芯片资源和功耗约束将算法改造成 ASIC 芯片可实现的定点形式，将算法细化为芯片上的基本运算并考虑定点量化噪声的影响。

对于问题一，给定波特率为 150Fbaud 的标准 16QAM 信号，固定线宽及色散值下，不考虑定点量化，以 RSNR 代价小于 0.3dB 为目标，设计性能最优的 CR 算法，使 Pilot 开销最小。首先需要搭建简易数字通信系统，实现 16QAM 信号的调制解调，并加入高斯白噪声观察星座图变化，而后尝试加入色散并对其进行色散补偿，最后进行 CR 算法的设计。在 CR 算法设计中，首先需要确定导频的插入的方式，对导频在信道中的相噪进行计算，而后需要确定插值的对象相噪而非相位补偿，最后遍历相邻导频的间隔来观察通信系统的误码率，在符合误码率要求的相邻导频间隔中取数值最大为 2900，此时 Pilot 开销最小，约为 $3.45e-4$ 。导频相位计算过程中会出现毛刺，干扰系统误码率判断，此时要在插值获得传输信号相噪之前，对导频相噪进行判断将毛刺去除，提升插值方法的准确度，进而降低误码率。

对于问题二，基于问题一已经设计好的 CR 算法模型，此时线宽与色散都是变化的场景，依然以 RSNR 小于 0.3dB 为目标，定量挖掘色散与线宽对于 Pilot 开销的影响。为了方便观察两个因素对于 Pilot 开销的影响，我们选择固定一个因素，遍历另外一个因素和相邻 Pilot 间隔数值，后两者与系统误码率拟合为三维曲面，并得到拟合方程以及方程系数。经过对比可以发现，色散与相噪存在耦合效应。当线宽较小时，相噪较小，色散补偿能够起到较大的作用，通过导频插值可以较好地补偿相噪；当色散较小时，色散与线宽的耦合对相噪的影响都不大，减小导频间隔，增大 Pilot 开销就能够很好补偿相噪。当色散和相噪都很大时，色散补偿作用有限，补偿后的相噪仍有混叠，插值效果不佳，难以达到 BER 门限。

对于问题三，是在问题二的场景上，将芯片实现资源纳入考察，需要考虑定点量化对性能和资源的影响，且导频开销可以任意变化，设计资源最低的 CR 算法。此时实际 Pilot 插入方式为每 128 个信号为一组，前 127 个信号为有效信号，第 128 个信号为导频。整体

框图设计分为软件部分与芯片部分：软件部分仅实现对色散补偿后的信号进行定点量化处理，其中包括对色散补偿后的信号进行归一化处理，以及在保证性能的条件下寻找最小位宽，实现芯片资源占用最小的目标；芯片部分包括四个子模块，分别为用改进的 LUT 查表法获得导频的相位 θ_a 、用线性插值法获得每个有效信号的相位、用查表法获得有效信号需要补偿的相位 $e^{-j\theta}$ 、对有效信号进行相位补偿。通过定点量化获得位宽与误码率的关系曲线并挑选满足误码率门限的最小位宽为 10 位，一位符号位，两位整数位，七位小数位。在芯片部分实现中给出了逻辑图以及计算资源，最后给出本问题的两个算法优化设计分析，一个是位宽优化，一个是查表优化。

对于问题四，基于问题三选定场景为 $Dz=5000\text{ps/nm}$ ， $LW=100\text{kHz}$ ，在问题三的算法中增加相邻 Pilot 间隔数值遍历获得误码率门限下的位宽与导频间隔组合，根据数据发现相同的位宽下，较小的导频间隔之间差距不明显，较大的导频间隔会使误码率提升，系统性能下降；而在相同导频间隔下位宽越大，性能越好，由此折衷采用较小的位宽和较大的导频间隔。基于误码率、位宽、导频间隔可以构造代价函数，在限定的误码率之下，代价函数与导频间隔的 \log 值成反比，基于三次拟合可以获得导频间隔、代价函数与位宽之间的方程，此时系统可以根据可承受代价的大小以及相邻 Pilot 间隔来自动选取对应位宽。

关键字： CR 算法 ASIC 芯片 相噪 Pilot 定点量化

目录

1. 问题重述	4
1.1 引言	4
1.2 问题的提出	4
1.2.1 问题一——最优 CR 算法设计	4
1.2.2 问题二——定量挖掘色散、线宽对 CR 算法性能影响	6
1.2.3 问题三——资源消耗最低的 CR 算法设计	6
1.2.4 问题四——性能、资源权衡下的 CR 算法设计	6
2. 模型的假设	7
3. 符号说明	8
4. 问题分析	10
4.1 问题一分析	10
4.1.1 子问题一——简易数字通信系统搭建	10
4.1.2 子问题二——系统色散补偿	13
4.1.3 子问题三——CR 算法消除相噪	14
4.1.4 最优 CR 算法实现	15
4.2 问题二分析	17
4.2.1 子问题一——固定 LW 下 Dz 对 Pilot 开销的影响	17
4.2.2 子问题二——固定 Dz 下 LW 对 Pilot 开销的影响	21
4.2.3 问题二结论	23
4.3 问题三分析	24
4.3.1 算法框架设计	24
4.3.2 软件模块实现	25
4.3.3 芯片模块实现	26
4.3.4 优化分析	29
4.4 问题四分析	30
参考文献	33
附录 A 我的 MATLAB 源程序	34
1.1 性能最优 CR 算法设计	34
1.2 资源消耗最少 CR 算法设计	36

1. 问题重述

1.1 引言

基于 ASIC 实现的光数字信号处理（DSP）芯片被广泛运用于消费电子类产品中，是光传输领域的核心。7nm 芯片工艺制造的光传输芯片容量达到 800Gbps，相当于单光纤可实现 48Tbps 的容量，为近年来网络流量的爆发式增长保驾护航，但与此同时，该技术的垄断也使得我国科技行业发展在中美关系日趋紧张背景下饱受芯片断供之苦。近年来，在国家政策的扶植下，大量企业与高校实验室投入芯片研发的流程中，自晶圆厂、封装厂的建造到芯片算法的研究，资金以及人才争先涌入，中国正在努力挣脱西方世界对于国内芯片市场的控制与影响。

在 ASIC 芯片的 DSP 算法设计中，一般需要两个主要步骤，一是根据信道损伤的物理模型设计补偿算法，此时仅考虑浮点计算；第二部根据芯片资源和功耗约束，将算法改造成 ASIC 芯片可实现的定点形式，此时需要将算法细化为芯片上的基本运算，并考虑定点量化噪声的影响。如何在性能与资源间权衡，在具体场景下实现最优设计，是 DSP 芯片算法设计中的核心问题，在有限的资源下，综合考虑算法性能与芯片资源，最大化芯片价值。

1.2 问题的提出

1.2.1 问题一——最优 CR 算法设计

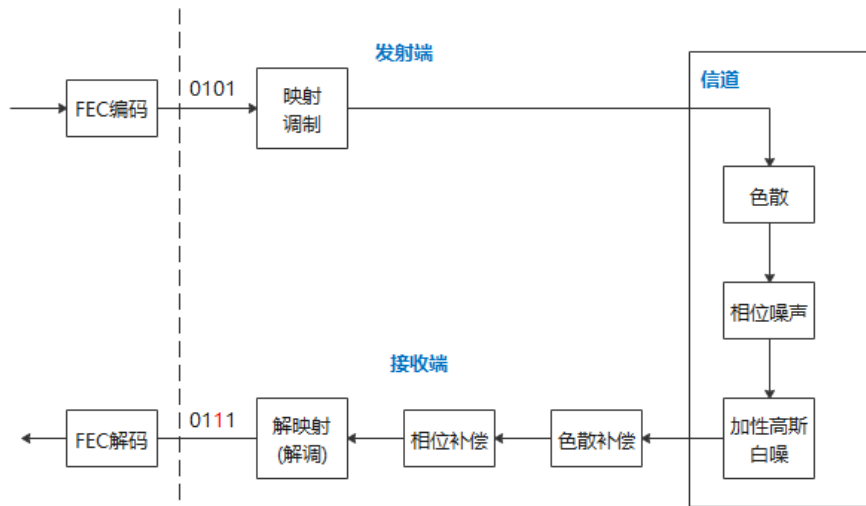


图 1 简化数字通信系统性能评估

本题考虑一个简化的数字通信系统性能评估模型，如图 1 所示。发送端编码后的二进制序列映射调制为星座点上的符号并向外发送，每秒发送的符号称为波特率 fBaud。信号在信道中收到色散和相位噪声的影响，同时受到加性高斯白噪声的影响，因此接收端在接收到信号后需要先补偿色散，再由载波恢复（Carrier Recovery, CR）算法补偿相位噪声，对

信号进行判决逆映射为二进制比特序列。典型相位噪声演化曲线如图 2 所示，由于信道中的色散与噪声干扰，星座图会发送扩散，从而导致信号判错引入误码，错误比特占总比特的比率称为误码率（BER）。

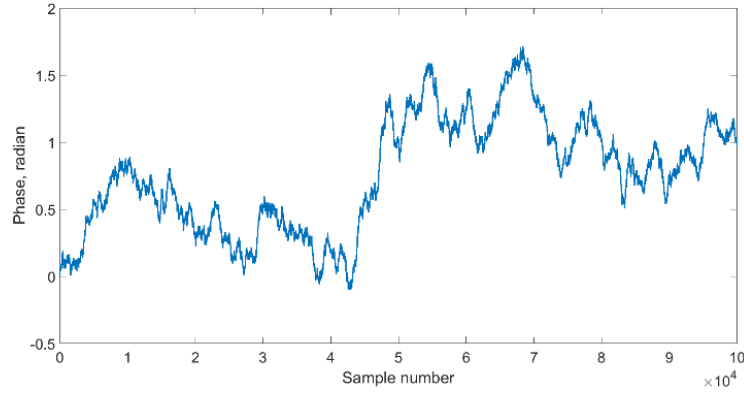


图 2 典型相位噪声演化曲线

算法评估中常用 RSNR（Required SNR）代价来评估算法性能，在只有加性高斯白噪声存在时，对于给定的调制格式，SNR 和 BER 之间的关系是确定的，将 BER 达到给定门限时信道中增加的 SNR 定义为 RSNR，相当于此时系统中能够容忍的噪声量。由于存在相位噪声以及色散干扰，相同门限点对应的 SNR 值会提高，相应系统中能够容忍的噪声量也相应减少，将 RSNR 的增加部分称为 RSNR 代价，是衡量系统的算法性能的常用指标。CR 算法性能越好，那么 RSNR 代价也应该越低。

典型的 CR 算法是间隔性地插入已知的导频符号（Pilot），通过比较接受信号和已知符号的相位差，来估算当前的相位噪声，再将此相位差反乘到接收端受影响的符号上，以实现相位噪声的补偿。Pilot 占总符号的比例称为 Pilot 开销，在 CR 算法设计中，由于 Pilot 本身为已知量并不传递信息，应尽可能减小 Pilot 占比来降低系统开销。

对于 CR 算法设计，文中提出以下问题，此时已有条件波特率为 150Gbaud 的标准 16QAM 信号，线宽为 100kHz，色散为 2 万 ps/nm，算法并行度固定为 128，仅考虑算法性能，不考虑定点量化，目标为在 RSNR 代价小于 0.3dB 的情况下，设计 CR 算法使得 Pilot 开销最小。

综上，问题一需要解决以下问题：

- （1）简易数字通信系统搭建实现信号调制、星座图与误码率 BER 计算；
- （2）针对色散效应对系统进行色散补偿；
- （3）设计 CR 算法抑制相位噪声对系统的影响并达到最优。

1.2.2 问题二——定量挖掘色散、线宽对 CR 算法性能影响

基于问题一模型，考虑线宽从 10kHz 10MHz，色散从 0 10000ps/nm 变化场景，在 RSNR 代价依然小于 0.3dB 的情况下，探究 Pilot 开销与线宽与色散间的关系，分析线宽与色散对于 CR 算法模型性能的影响。该问题可以拆解为以下两个子问题：

- (1) 固定线宽下的色散变化场景，探究 BER 门限下 Pilot 开销与色散间的关系；
- (2) 固定色散下的线宽变化场景，探究 BER 门限下 Pilot 开销与线宽间的关系。

1.2.3 问题三——资源消耗最低的 CR 算法设计

通常在计算机上开展数学计算一般基于软件，而在 ASIC 上的计算往往基于硬件电路，因而如果要计算某个公式，通用计算机上是转换为一条条逻辑指令，按时间顺序在 CPU 中运行并最终输出结果。而在 ASIC 上开展的计算，则是把计算过程拆分为具体的加法、乘法等基本操作，每个基本操作都对应了不同的专用逻辑电路，在芯片上各自占据了一定的面积，并且在使用中占据一定的拍数。由于芯片上的计算都是在系统时钟下以一拍一拍的节奏运行，如果采用串行处理的方式会导致芯片时钟频率极高，而且使得芯片的功耗几何级数增长，所以想要增加处理流量，就需要考虑并行处理方法用资源换取处理流量。此外，计算机中常用双精度浮点数来定义参变量，而 ASIC 通常用定点数来表示参变量的大小，由于 ASIC 定点数位数往往较小，舍入误差增大，会引入量化噪声，由于 ASIC 中任何计算都应当量化为定点数的计算，量化噪声的影响也是算法设计的关键之一。

综上设计 ASIC 芯片上的 DSP 算法，除了考虑算法原本的性能外，还需要考虑并行实现、定点量化、时序约束和资源以及功耗等约束。本题在设计 CR 算法过程中忽略算法性能，将芯片实现的资源纳入考量，此时需要考虑定点量化对于性能和资源的影响，该情况下导频开销可以任意变化，但静荷的流量大于 145GBaud，目标在于设计资源消耗最低的 CR 算法。因此本问题可以分为以下几个子问题：

- (1) 算法框架设计；
- (2) 软件模块实现；
- (3) 芯片模块实现。

1.2.4 问题四——性能、资源权衡下的 CR 算法设计

在获得最优性能的 CR 算法与最小资源消耗的 CR 算法后，需要考虑现实应用中，算法性能与芯片实现可利用资源二者需要进行权衡，选出问题三中最有代表性的一个场景，给出统筹“性能——资源”综合考虑下的算法设计思路，构造性能和资源的综合代价函数，给出一套自动优化位宽和实现性设计的方案，并给出定量结果。

2. 模型的假设

根据题意，我们做出如下假设：

- 本题中不考虑纠错编码，BER 均指直接判决后的 BER。
- 本题中 BER 门限值设计为 $2\text{E-}2$ ，RSNR 代价为 0.3dB。
- 本题中色散补偿按照给定理想方式进行计算。
- 本题中 FPGA 是可定制的，不用限制位数。
- 本题中，ASIC 芯片 1 个时钟周期最多完成 1 级乘法，4 级加法，以及 1 级查表操作。
- 本题中，并行度 128 指可以同时进行 128 个符号的计算，每一拍输入 128 个符号，输出 128 个运算后的符号。
- 本题中，8+8bit 复数加法器占用资源为 1U，则设定 8+8bit 实数加法器占用资源为 0.5U。
- 本题设计均不考虑色散补偿和误码率计算的复杂度和资源，只考虑 CR 算法的相关资源，其中包含计算相噪和补偿相噪。

3. 符号说明

本题涉及符号在下表 1 表 2 列出。表一涉及问题一与问题二相关符号，表二涉及问题三与问题四相关符号。

表 1 问题一及问题二相关符号

符号	意义
K	信号比特个数
r_k	理想星座点
s_k	接收到的符号
n_k	噪声
P_n	噪声的平均功率
P_s	信号的平均功率
N	总传输的符号数
SNR	信噪比（dB）
P_c	16QAM 系统的正确判决概率
P_4	四元 PAM 的错误概率
P_e	16QAM 系统的错误概率
λ	波长（1550nm）
Dz	色散值
c	光速（ $3 * 10^8 \text{m/s}$ ）
f	频点
f_b	波特率
H(f)	色散效应
FFT	快速傅里叶变换
IFFT	傅里叶反变换
exp	e 的指数函数
CR	载波恢复
t	某一时刻
$S_0(t)$	叠加相噪前的波形
$S_1(t)$	叠加相噪后的波形
$\theta(t)$	相噪
$d\theta$	相位差
LW	激光器线宽指标（kHz）

X_k	均值为 0，方差为 1 的随机变量
Pilot	导频符号
gap	相邻导频的间隔符号数
SSE	和方差
R-square	确定系数
Adjusted R-square	校正确定系数
RMSE	均方根

4. 问题分析

4.1 问题一分析

4.1.1 子问题一——简易数字通信系统搭建

二进制序列通常需要将 K 个比特作为一个“符号”进行传输，每个符号有 $2K$ 个不同状态，光传输利用光波的复振幅承载信号，因此可用复平面上不同的点来对应不同的符号状态，这种将符号状态画在复平面上的图称为“星座图”，图上的点称为“星座点”。如图 3（a）所示 QPSK 调制，经过信道叠加噪声和接收机处理后，接收端的星座图就不再是理想的四个点，而是会出现扩散。当接收机收到 1 个符号时，就将发送的符号判定为离该符号最近的星座点，如果噪声过大就会出现判错的可能，如图 3（b）中所示的蓝点。误码率 BER 定义为错误的比特数占总传输比特数的比例，是衡量通信系统性能的最根本指标，也是本题求解的重要条件约束之一。

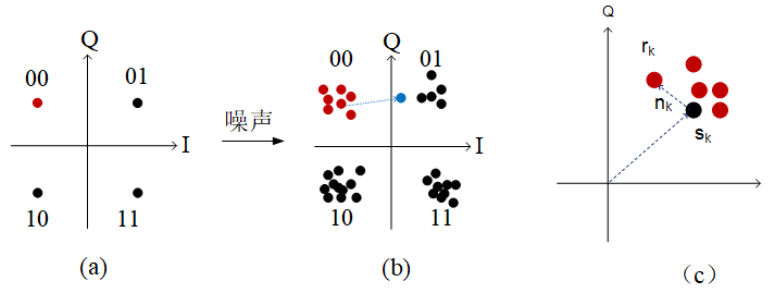


图 3 星座图与噪声导致误码的示意图及信号和噪声定义图

图 3（c）中的理想星座点用 s_k 表示，接收到的符号用 r_k 表示，则噪声为

$$n_k = r_k - s_k \quad (1)$$

噪声通常服从均值为 0 的正态分布，噪声的方差等于噪声的平均功率，定义为

$$P_n = \frac{1}{N} \sum_{k=1}^N |n_k|^2 \quad (2)$$

其中 N 为总共传输的符号数，信号平均功率定义为发送符号绝对值平方的均值：

$$P_s = \frac{1}{N} \sum_{k=1}^N |s_k|^2 \quad (3)$$

定义信号和噪声功率的比值为信噪比 SNR，

$$SNR = \frac{P_s}{P_n} \quad (4)$$

工程上通常用 dB 作为 SNR 的单位，定义为

$$SNR(dB) = 10 \log_{10} \frac{P_s}{P_n} \quad (5)$$

QAM 调制为正交幅度调制技术，在 QAM 调制中，数据信号由相互正交的两个载波的幅度变化表示，QAM 是一种矢量调制，将输入先使用格雷码映射到一个复平面上，也就是我们所看到的星座图，形成复数调制符号，然后将该符号的 I、Q 分量（即该附属的实部和虚部）采用幅度调制，分别对应调制在相互正交的两个载波上。本题通信系统搭建使用标准 16QAM 信号，其在星座图上具有 16 个样点，每个样点表示一种矢量状态，16QAM 有 16 态，每 4 位二进制数规定了 16 态中的一态，16QAM 的每个符号时间传送 4 比特映射。根据 802.11a 规定，基于格雷码，16QAM 编码表如表 2 所示。

表 2 16QAM 编码表

输入 (b0b1)	I	输入 (b2b3)	Q
00	-3	00	3
01	-1	01	-1
11	1	11	1
10	3	10	-3

本题对于随机信号 01 序列进行 16QAM 调制，在没有噪声干扰的情况下，使用 Matlab 仿真长度为 10^7 的随机 01 信号，对其进行 16QAM 调制得星座图如图 4 所示。

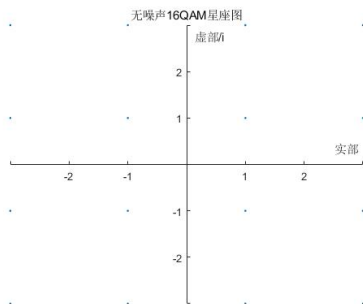


图 4 无噪声的 16QAM 信号星座图

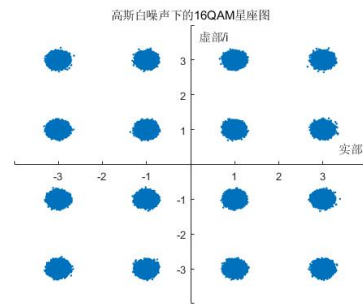


图 5 SNR=100 的 16QAM 信号星座图

图 4 中，16 个落点均匀分布在坐标轴上，分别代表信号的 16 个状态，可理解为没有经过信道的调制信号。由于信号在经过信道时，势必会受到加性高斯白噪声的干扰，此时接收端收到的星座图不再是 16 个点，信号会散落在 16 个状态点周围，信噪比越小，落点

越分散。为了描述高斯白噪声对于接收端信号落点的影响，对 SNR 为 100 时对信道进行仿真，得星座图如图 5 所示。

由于本题已知 BER 门限为 0.02，在仅存在高斯白噪声的情况下，SNR 与 BER 存在得关系确定，可以通过 BER 数据逆推出符合该 BER 门限系统 SNR，用于计算后续 RSNR 代价。对于 16QAM 来说，它的信号矢量端点分布图和两个正交载波上两个 PAM 信号的矢量端点分布等效，其中每一个具有 4 个信号点，所以 16QAM 系统的正确判决概率是：

$$P_c = (1 - P_4)^2 \quad (6)$$

该式中， P_4 是 4 元 PAM 的错误概率，在等效 QAM 系统的每一个正交信号中，4 元 PAM 具有一半的平均功率，通过适当的修改 4 元 PAM 的错误概率，可以得到

$$P_4 = 2(1 - \frac{1}{4})Q(\sqrt{\frac{3}{15}}SNR^2) \quad (7)$$

已知 Q 函数为

$$Q(a) = \frac{1}{2}erfc(\frac{a}{\sqrt{2}}) \quad (8)$$

此时 16QAM 的错误概率为：

$$P_e = 1 - (1 - P_4)^2 \quad (9)$$

基于上述公式，通过 Matlab 仿真得到在仅存在高斯白噪声的环境下，16QAM 信号在传输过程中 SNR 与 BER 的关系曲线，具体见图 6 所示。理想状况下，逼近 BER 为 0.02 门限的 SNR 值约在 5.44dB 左右，而使用高斯白噪声对信号进行仿真时发现，逼近 BER 为 0.02 门限的 SNR 值约在 5.72dB 左右。使用 Matlab 自带的 awgn 函数对 16QAM 调制后的数据进行噪声叠加，获得该噪声下的 16QAM 星座图，如图 7 所示，可以看到数据散落已无法准确从星座图中分出 16 个点。

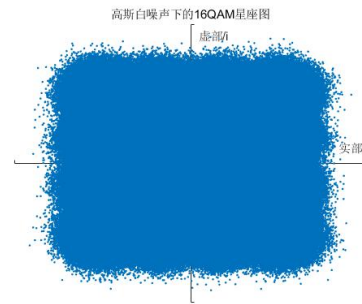
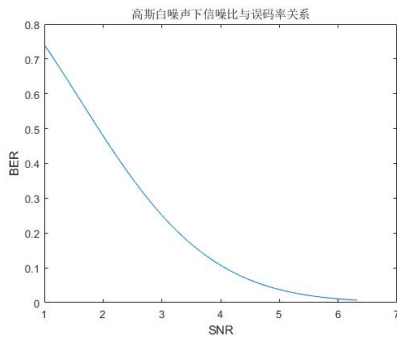


图 6 高斯白噪声下信噪比与误码率关系 图 7 符合 BER 门限 16QAM 信号星座图

4.1.2 子问题二——系统色散补偿

光纤中色散的效应，可认为是对信号直接做傅里叶变换后的频域数据施加 1 个随频点平方关系变化的相位，如下式所示。

$$H(f) = \exp[j * (\frac{\lambda^2 \pi D z}{c} f^2)] \quad (10)$$

其中， λ 是波长，取 1550nm， Dz 是色散值， c 是光速， f 为频点，本题施加的色散与算法中补偿色散如图 8 所示，假设色散值已知，则先 FFT 将接收到的数据转换到频域，再乘上式中的频域相应，随后再 IFFT 转换回始于即可。信道中色散的频域响应和算法中补偿色散的频域响应呈共轭关系。

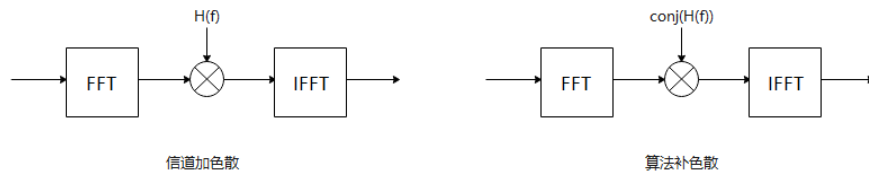


图 8 色散和色散补偿方法

根据上述公式，在子问题一搭建的简易通信系统中加入色散以及色散补偿，此时考虑本题中提及的 RSNR 代价小于 0.3dB，将 SNR 设定为子问题一求得的 SNR 值与 0.3dB 相加。此时获得在加性白高斯噪声环境中加入色散的 16QAM 星座图与色散补偿后的 16QAM 星座图，分别见图 9、图 10。

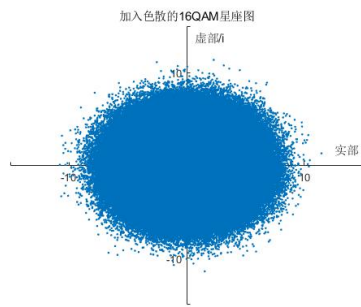


图 9 加入色散的 16QAM 星座图

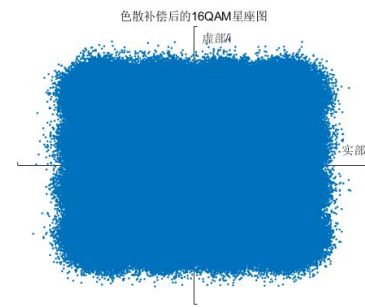


图 10 色散补偿后的 16QAM 星座图

图 9 与图 7 比较，可以发现加入色散以后，原有的矩形星座图已无法识别边缘指教，误码率高达 0.5，随后对信号进行色散补偿后，误码率重新降回 0.02 以下，星座图恢复矩形轮廓，仅受高斯白噪声影响。

4.1.3 子问题三——CR 算法消除相噪

在通信系统中，相噪会对信号叠加一个时变的相位，

$$S_1(t) = S_0(t) \exp\{j\theta(t)\} \quad (11)$$

其中， $S_0(t)$ 是叠加相噪前的波形， $S_1(t)$ 是叠加相噪后的波形， $\theta(t)$ 是相噪。通信系统常用等间隔采样后的离散方式表示，每个采样点对应了某个时刻采样到的波形。对应于相位噪声， $k+1$ 时刻的相位差表示为：

$$d\theta = \theta_{k+1} - \theta_k = \left(\sqrt{\frac{2\pi * LW}{f_b}} \right) * X_k \quad (12)$$

其中， LW 是一个激光器线宽指标，单位为 kHz 。 f_b 是波特率，而 X_k 是均值为 0，方差为 1 的随机变量。

典型的 CR 算法是间隔性地插入已知的导频符号（Pilot），如图 11 所示。通过比较接收信号和已知符号的相位差，来估算出当前的相位噪声，再将此相位差反乘到接收端受影响的符号上，就实现了相位噪声的补偿。Pilot 占总符号的比例称为 Pilot 开销。一般在设计 CR 算法时，由于 Pilot 本身为已知量并不传递信息，应当尽可能减小 Pilot 占比来降低系统开销，也是本题的设计目标。此外，由于加性白噪声会影响相位估计精度，可以用两个连续的 Pilot 符号求平均以抑制白噪声影响，也可以在间插的 Pilot 之间求平均。

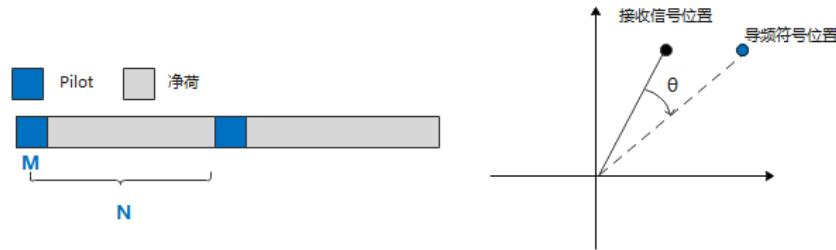


图 11 Pilot 的载波恢复算法

插值是根据已知的数据序列，找到其中的规律，而后根据找到的规律对其中尚未有数据记录的点进行数值估计。基于这样一种功能，它可以对数据中的缺失进行合理的补偿，在 CR 算法设计中，我们可以对已知 pilot 的相位偏差进行计算，并利用插值对信号序列的相偏进行合理补偿，最终处理得到去除相噪的信号。本题中，我们所采用的插值方法是线性插值，这是一种针对一维数据的插值方法，它根据一维数据序列中需要插值的点的左右邻近两个数据点来进行数值的估计，但并非简单求解这两个点数据大小的平均值，而是根据到这两个点的距离来分配它们的比重。已知点 (x_0, y_0) 、 (x_1, y_1) ，在 x 处插值，可以通过计算得到 y ，具体见下式。

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \quad (13)$$

本题中所使用的一维线性插值方法利用 Matlab 自带的 interp1 函数实现。

根据上述方法,对 16QAM 通信系统加入相噪并对其进行相位补偿,加入相噪的 16QAM 星座图见图 12, 相位补偿后的 16QAM 星座图见图 13。

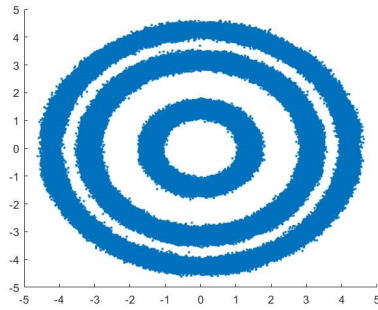


图 12 加入相噪的 16QAM 星座图

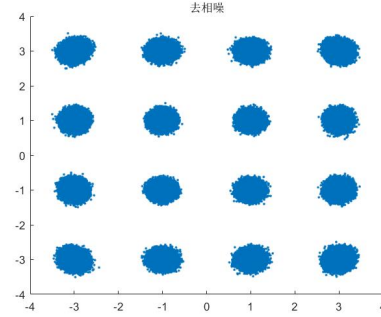


图 13 相位补偿后的 16QAM 星座图

4.1.4 最优 CR 算法实现

基于对上述三个子问题的分析,搭建 Pilot 开销尽可能小的 CR 算法,依照题目给定的 150Gbaud 波特率标准 16QAM 信号,线宽为 100kHz,色散为 20000ps/nm,算法并行度固定为 128,不考虑定点量化,以 RSNR 小于 0.3dB 为目标设计 CR 算法,使得 Pilot 开销最小,具体代码见附录 A。由于本题不考虑 ASIC 芯片资源的问题,所以并未强制将 Pilot 插入限定在每 128 个符号一组,而是放开导频与导频间的间隔符号数 gap, gap 取值从 0 至 5000,每一个固定的 gap 值对应的误码率为每隔该距离均匀插入一个 Pilot 用于载波恢复最终所得的误码率。本问题导频插入方法见图 14,均匀插入导频,且当最后一串数据结束后,无论跟上一个导频有多远,都在末尾添加一个导频以恢复该段数据。



图 14 导频插入方法

在 CR 算法设计中,插值对象设定为相噪,而非受相噪干扰后的数据,原因在于相噪是高斯白噪声的不断累加,而相噪作用于信号后的数据经过指数计算,因而对相噪进行插值可以直接使用一维线性插值,准确度相对较高。

CR 算法的流程框图如图 15 所示。

在搭建算法模型的过程中发现,由于相位的周期是 2π ,在相噪恢复的过程中会出现相

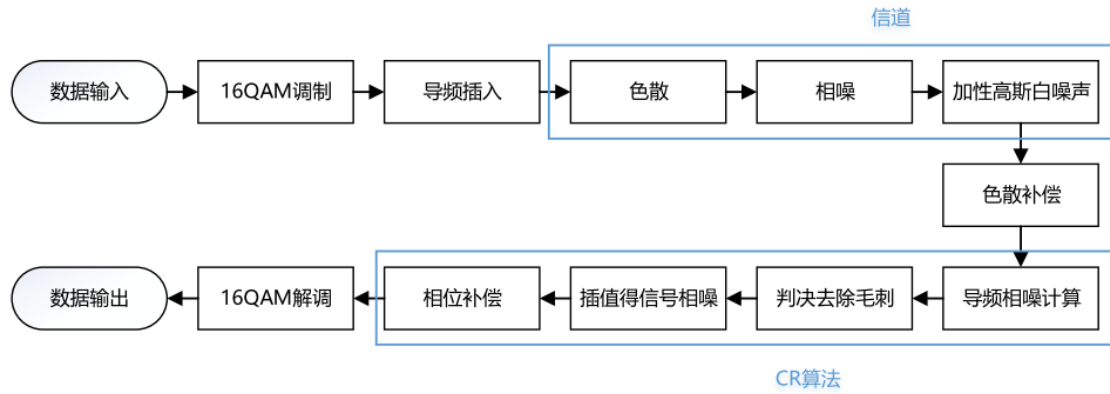


图 15 流程图

位误判导致毛刺出现，见图 16，而导频相位的错误判断则会直接导致插值得到的全部数据的相位出现毛刺，见图 17。

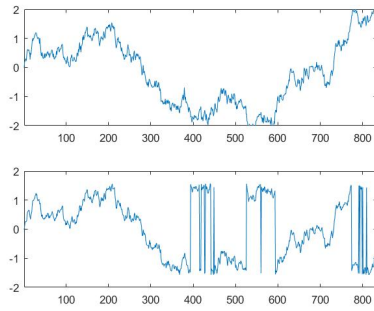


图 16 导频相位与插值恢复导频相位

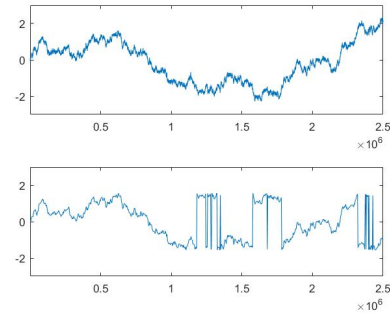


图 17 数据相位与插值恢复数据相位

为了解决这个问题，此时选择在相位恢复后添加判决条件，如果相邻前后两个相位差值超过 1.5π ，则认为出现误判，并基于前一个相位的高低相应选择在后一个相位上加 2π 或者减 2π ，将出现误判导致的毛刺消除。

在仿真中还出现以下情况，如果将插入 Pilot 的幅值提升可以在一定程度上提升系统性能降低误码率，但是在后续的设计中，由于需要考虑 ASIC 资源，Pilot 的功率增大会导致资源消耗过大，因此在本题中尚考虑将 Pilot 幅值取到对系统性能提升最优的情况，但后续将考虑尽可能降低 Pilot 幅值。为了能够找到本题限定条件下较为适合的 Pilot 的幅值，我们遍历 Pilot 幅值取值 1 至 50，并设置 Pilot 间隔 100 至 5000 个符号，按照 100 个间隔递增，进行双重循环，找出符合 BER 门限小于等于 0.02 的组合，最后将挑选出的组合拟合出曲线如图 18 所示，为了能够取到较高的相邻 pilot 的间隔，此时 Pilot 的开销最小，取幅值为 24。

在确定幅值之后，将相邻 Pilot 间隔 gap 视为自变量，单位为 1k，系统误码率视为因变量，拟合得到误码率与 gap 之间的关系曲线如图 19 所示。由图 19 分析出相邻 Pilot 间隔与

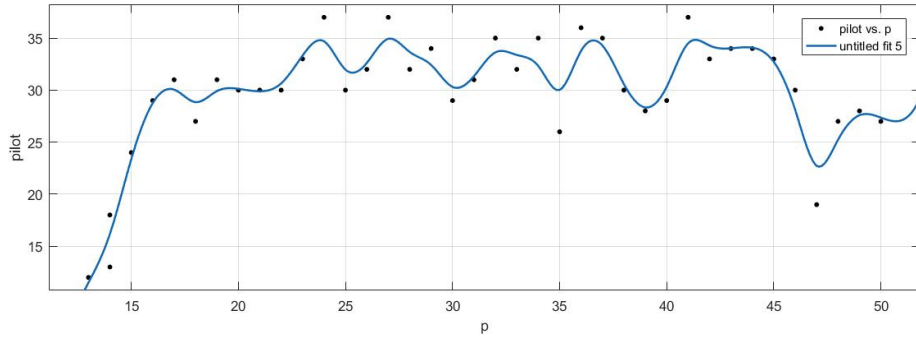


图 18 Pilot 间隔与其合适幅值的关系曲线

误码率之间的关系并不是线性的，而是有一定的符合条件的范围值。在 Pilot 间隔较小的情况下，系统误码率不符合门限要求，在 Pilot 间隔位于 800 至 2900 之间的情况下，BER 能够较为稳定符合 0.02 门限值，出于对于 Pilot 开销尽可能小的考虑，我们选取 Pilot 间隔在 2900，即每隔 2900 个符号插入一个 Pilot，由于信号长度随机，如果信号的最后一位并没有能够正好在其后插入 Pilot，则自动补上一个 Pilot，此时 Pilot 的开销约为 $3.45E-4$ 。

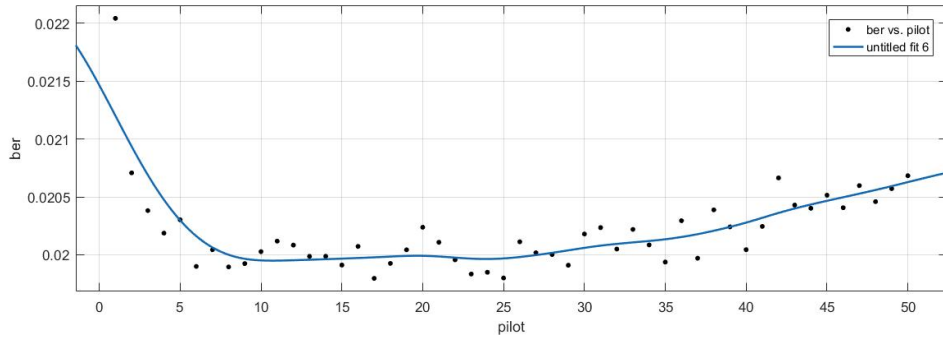


图 19 BER 与 Pilot 间隔的关系曲线

4.2 问题二分析

本题考虑线宽 LW 从 10kHz 至 10MHz，色散 D 从 0 至 10000ps/nm 变化的场景，依然以 RSNR 小于 0.3dB 为目标，定量挖掘这两个因素对于 Pilot 开销的影响，系统搭建基于题目一已完成的模型。

4.2.1 子问题一——固定 LW 下 Dz 对 Pilot 开销的影响

为了能够更好观察单一因素对 Pilot 开销的影响，我们考虑先分析线宽 LW 数值一定的情况下，色散从 0 至 10000ps/nm 的变化场景。当 LW 取 100kHz 时，gap 从 100 至 5000 以 100 为步长遍历，色散、Pilot 间隔以及误码率的关系见图 20 所示，该图使用 Matlab 的 Curve Fitting Tool 工具箱对三项数据进行拟合。从图中可以发现，在 LW 线宽一定的情况下，色散越大，Pilot 间隔符号数量越大（Pilot 开销越小），误码率越高。但是 Pilot 间隔不

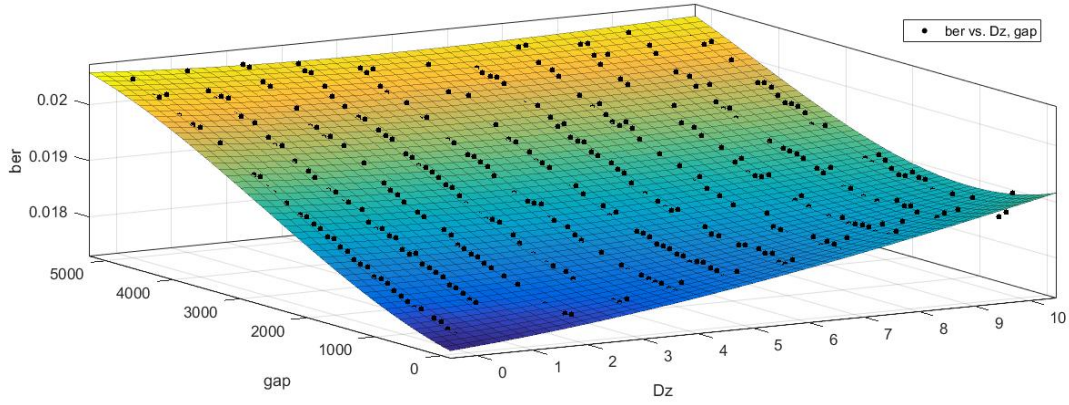


图 20 LW = 100kHz 下的 Dz、Pilot 间隔与误码率关系图

断变小（Pilot 开销变大）并非一定能够提升系统性能，由此在 Pilot 间隔趋于 0 的情况下，我们可以看到误码率会提升，考虑可能的原因为 Pilot 开销不断变大后，线宽不为零导致相噪一定存在，同是又存在环境中的高斯白噪声。基于算法我们可以将相噪作用于信号时的式子写为

$$S_1(t) = S_0(t)\exp\{j\theta(t)\} + a + bi = S_0(t)(\exp\{j\theta(t)\} + a_1 + b_1i) \quad (14)$$

其中 $a + bi$ 为高斯白噪声，我们在对信号进行相位补偿时考虑信号的噪声构成，所以相位补偿会同时作用于高斯白噪声，因为相噪是基于传送数据个数不断累加的，由 Pilot 开销变大后系统内 Pilot 个数提升，所以在信号序列较为靠前的地方就已经开始插入 pilot 并且十分密集，根据图 2 中典型相位噪声演化曲线可以发现前期的相位噪声非常小，后期才逐渐增大，那么在 $S_1(t)$ 序列中，相噪是在随着时间增大，而高斯白噪声不变，因而在前期使用较小的相噪影响下的 Pilot 进行相噪计算势必会严重受到高斯白噪声的干扰，导致相位补偿受到偏差，进而导致误码率提升。由此我们可以知道，增加 Pilot 开销确实能够提升估算噪声的准确度，但是同时也会带来基于信号前部序列由于相位噪声较小导致的基于高斯白噪声进行的相位补偿，由此增加相位的错误估计从而带来误码率，由于一般情况下我们寻求较低的 Pilot 开销，所以在实际应用中可以暂时忽略高斯白噪声给相噪补偿带来的影响。

而当 gap 值较小，即趋近于 0 时，此时 Pilot 开销很大，可以明显看到固定 LW 下，色散越大，误码率越高。gap 值较小时，我们认为其能够较为准确估算相噪，可以忽略插值所带来的偏差。理论上色散补偿是可以完全消除色散影响的，当色散为 0 时，BER 最小，此时仅有相噪与高斯白噪声，而随着色散不断增大，色散与相噪耦合增强，而色散补偿作用有限，因此当 Dz 上升时，会出先明显的误码率上升，而不是一条与坐标轴平行的水平线。当 gap 不断增大，趋近于 5000 时，此时 Pilot 开销很小，插值估算相噪所带来的误差要高于色散耦合导致的影响，所以并没有看到曲面边沿随着 Dz 的增大而出现明显的误码率变化。

综上我们可以确认在符合 BER 门限的条件下，色散越小符合条件的可用 gap 越大，Pilot 开销越小。通过工具箱拟合三次曲面方程为：

$$f(x, y) = p_{00} + p_{10} * x + p_{01} * y + p_{20} * x^2 + p_{11} * x * y + p_{02} * y^2 + p_{30} * x^3 + p_{21} * x^2 * y + p_{12} * x * y^2 + p_{03} * y^3 \quad (15)$$

其中，x 为 Pilot 间隔 gap，y 为 Dz 取值，f(x,y) 为 BER，p 为方程系数，具体取值见表 3。

表 3 LW=100kHz 下的曲面拟合系数

系数	数值
p_{00}	0.02073
p_{10}	-1.587e-5
p_{01}	7.837e-9
p_{20}	1.055e-8
p_{11}	1.024e-12
p_{02}	-3.311e-16
p_{30}	-1.771e-12
p_{21}	7.317e-16
p_{12}	-1.19e-19
p_{03}	1.589e-23

此时，曲面的拟合优度如下：和方差 SSE 为 6.408e-06，确定系数 R-square 为 0.9785，校正确定系数 Adjusted R-square 为 0.9781，均方根 RMSE 为 0.0001089。由于 SSE 越接近于 0，表明模型选择和拟合越好，数据预测也越成功，显然本模型符合该条件。

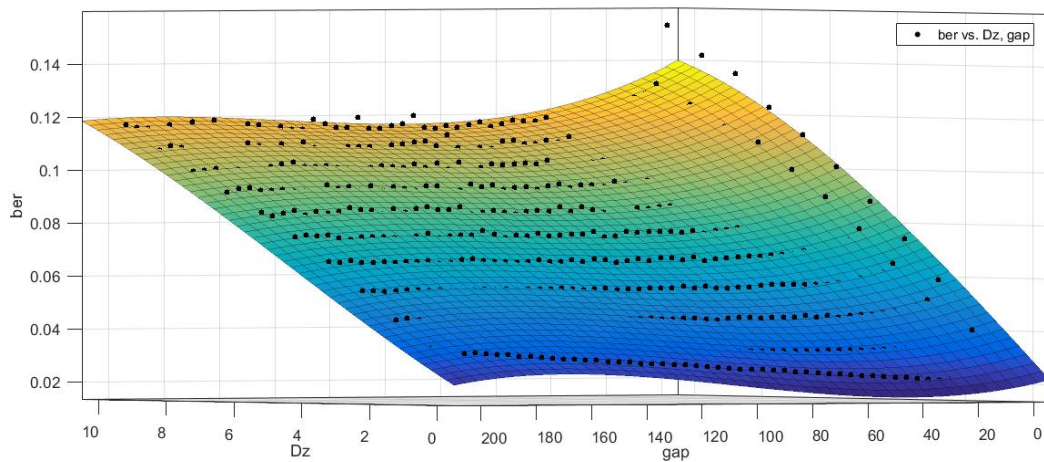


图 21 LW = 10MHz 下的 Dz、Pilot 间隔与误码率关系图

当 LW 取值为 10MHz 时，gap 从 4 至 200 以 4 为步长遍历，拟合曲面见图 21。此时曲面的拟合优度如下，SSE 为 0.003935，R-square 为 0.992，Adjusted R-square 为 0.9919，RMSE 为 0.0027。

分析图 21 中 Dz 与 gap 间的关系，可以发现变化趋势与 LW 取 10kHz 时一致，Dz 数值越小，能够容纳的 gap 取值越大，Pilot 开销越少，但是 z 轴误码率坐标几乎全部超出了 BER 为 0.02 的门限，Pilot 开销越大但还未趋于 0 时，理论上 BER 不断下降，但此时 BER 也没有低于 0.02 门限，由此也可得判断出 LW 取值越大会使得 Pilot 开销变大，甚至 Pilot 开销变大也没有办法符合系统误码率要求。此时拟合方程见式（15），拟合系数见表 4。

表 4 LW=100kHz 下的曲面拟合系数

系数	数值
p_{00}	0.02719
p_{10}	0.01351
p_{01}	-0.0002536
p_{20}	-5.674e-05
p_{11}	-4.29e-05
p_{02}	3.306e-06
p_{30}	-2.238e-05
p_{21}	1.671e-06
p_{12}	8.519e-08
p_{03}	-1.018e-08

由于线宽给定范围值为 10kHz 至 10MHz，在本子问题分析中，取线宽值为 10kHz，20kHz，50kHz，100kHz，200kHz，500kHz，750kHz，1MHz，3MHz，5MHz，10MHz，分别进行曲面拟合，最终得到拟合优度数据表格见表 5。

表 5 不同 LW 下的曲面拟合优度

LW(Hz)	SSE	R-square	Adjusted R-square	RMSE
10k	2.636e-06	0.8243	0.8211	7.335e-05
20k	3.891e-06	0.928	0.9267	8.912e-05
50k	3.795e-06	0.9518	0.951	8.384e-05
100k	6.407e-06	0.9785	0.9781	0.0001089
200k	1.374e-05	0.9805	0.9802	0.0001595
500k	0.0002553	0.8634	0.8611	0.0006876
750k	4.991e-05	0.9901	0.99	0.000304

1M	0.0001226	0.9869	0.9867	0.0004764
3M	0.0008497	0.9887	0.9885	0.001254
5M	0.001679	0.9905	0.9903	0.001764
10M	0.003935	0.992	0.9919	0.0027

根据曲面拟合优度数据表可得，线宽不断增大的过程中，模型 SSE 不断变大，R-square 不断增大，RMSE 也在不断增大，由于已知 SSE 数值越小模型越越精准，分析是由于当 LW 增大时，误码率上限提升，数据跨度变大，导致落点更为分散，使得数据模型精准度下降。因而判断 LW 越小，模型精度越高，数据落点越集中，同等 Pilot 开销与 Dz 下模型误码率越小，因而在 BER 门限下能够容纳的 Pilot 间隔越大，Pilot 开销越小。

4.2.2 子问题二——固定 Dz 下 LW 对 Pilot 开销的影响

本题子问题一中确定 LW 固定时，Dz 越小，Pilot 间隔符号个数越大，Pilot 开销越小。此时考虑在固定 Dz 的情况下，LW 从 10kHz 至 10MHz 的变化场景。

本子问题中 gap 间隔设置为先从 5 至 40 以 5 为步长遍历，而后从 80 至 4000 以 80 为步长遍历，当 Dz 取值为 0 时，将 LW、gap 与 BER 三者关系拟合为平面图如图 22 所示。由图中分析可得，在 Dz 为 0 的情况下，LW 数值越小，在 BER 门限下能够容纳的 Pilot 间隔数据个数越大，Pilot 开销越小，符合子问题一分析中的猜想。但是在 gap 值接近于 0，Pilot 开销较大的情况下，LW 越小性能越差，与 Pilot 较小开销情况下的 LW-BER 关系相反，考虑原因可能是由于线宽所带来的相噪会通过 CR 算法恢复，但是导频位置的白噪声对相噪判决影响较大，导致误码率提升。拟合曲面方程见式 (15)，x 为 gap，y 为 LW，f(x,y) 为 BER，方程系数见表 6。

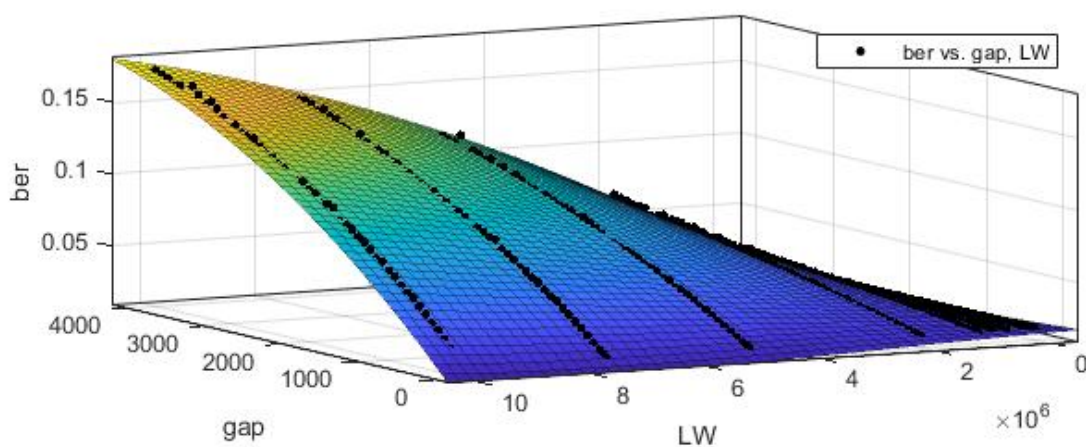


图 22 Dz = 0 下的 LW、Pilot 间隔与误码率关系图

表 6 LW=100kHz 下的曲面拟合系数

系数	Dz=0	Dz=5000ps/nm	Dz=10000ps/nm
p_{00}	0.01711	0.02223	0.0244
p_{10}	3.873e-07	-2.069e-05	-2.566e-05
p_{01}	6.15e-10	1.208e-08	2.19e-08
p_{20}	-4.212e-10	1.344e-08	1.592e-08
p_{11}	7.93e-12	-1.627e-12	-5.334e-12
p_{02}	-2.321e-16	-4.788e-16	-1.139e-15
p_{30}	1.047e-13	-2.23e-12	-2.583e-12
p_{21}	-5.082e-16	1.113e-15	1.453e-15
p_{12}	-2.127e-19	-6.777e-20	5.942e-20
p_{03}	1.995e-23	1.618e-23	3.28e-23

当 Dz 取值为 5000ps/nm 时，将 LW、gap 与 BER 三者关系拟合为平面图如图 23 所示，当 Dz 取值为 10000ps/nm 时，拟合平面图如图 24 所示。由图中可得 LW 值越小，符合 BER 门限的数据点越多，系统能够容纳的 gap 数值越大，Pilot 开销越小，与 Dz 取 0 值时的趋势一致。该情况拟合曲面方程见式 (14)，x 为 gap，y 为 LW，f(x,y) 为 BER，方程系数见表 6。由本子问题的曲面图比较可以发现，Dz 值固定而 LW 值在区域内变换，会出现较

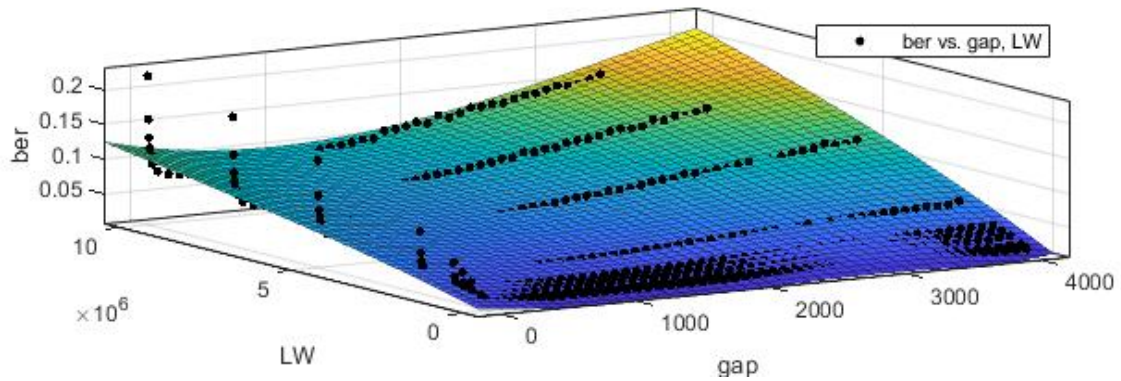


图 23 Dz = 5000ps/nm 下的 LW、Pilot 间隔与误码率关系图

高的误码率，仅有部分数据满足 BER 为 0.02 的门限要求，分析原因在于 LW 超过 $1e5$ 后，由于相噪过大，相位补偿难以将相噪完全去除，导致误码率上升。而通过 Dz 值设定为 0、5000、10000ps/nm 获得的三张曲面图进行对比，可以发现随着 Dz 增大，BER 上限值不断增大，这表示在 BER 门限下，能够容纳的 gap 值变小，Pilot 开销变大，呼应本题子问题一得出的结论，Dz 越小，Pilot 开销越小。

由于色散给定范围值为 0 至 10000ps/nm，在本子问题分析中，取色散值为 0，1000，

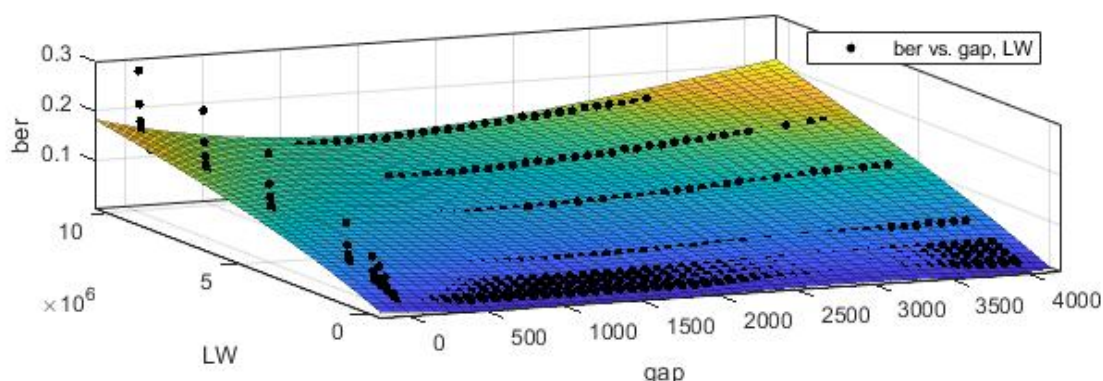


图 24 $Dz = 10000\text{ps/nm}$ 下的 LW、Pilot 间隔与误码率关系图

2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000ps/nm, 分别进行曲面拟合, 最终得到拟合优度数据表格见表 7。

表 7 不同 Dz 下的曲面拟合优度

$Dz(\text{ps/nm})$	SSE	R-square	Adjusted R-square	RMSE
0	0.0007277	0.9992	0.9992	0.000989
1000	0.01235	0.9853	0.9851	0.004075
2000	0.02911	0.9655	0.9651	0.006255
3000	0.04268	0.9519	0.9513	0.007574
4000	0.05468	0.9409	0.9401	0.008573
5000	0.06333	0.9362	0.9354	0.009226
6000	0.07286	0.9314	0.9306	0.009896
7000	0.07891	0.9311	0.9303	0.0103
8000	0.08706	0.929	0.9281	0.01082
9000	0.09144	0.9302	0.9293	0.01109
10000	0.09615	0.9319	0.931	0.01137

根据曲面拟合优度数据表可得, 色散不断增大的过程中, 模型 SSE 不断变大, R-square 不断增大, RMSE 也在不断增大, 由于已知 SSE 数值越小模型越越精准, 分析是由于当 Dz 增大时, 误码率上限提升, 数据跨度变大, 导致落点更为分散, 使得数据模型精准度下降。因而判断 Dz 越小, 模型精度越高, 数据落点越集中, 同等 Pilot 开销与 LW 下模型误码率越小, 因而在 BER 门限下能够容纳的 Pilot 间隔越大, Pilot 开销越小。

4.2.3 问题二结论

本题中，色散与相噪存在耦合效应，当线宽较小时，相噪较小，色散补偿能起到较大作用，通过导频插值可以较好地补偿相噪。当色散较小时，色散与线宽的耦合对相噪影响不大，减小导频间隔也能补偿较大的相噪。当色散和相噪都较大时，色散补偿的作用有限，色散补偿后的相噪仍有混叠，插值效果不佳，难以达到 BER 门限。

综上，在 RSNR 低于 0.3dB 的情况下定量分析色散、线宽与 Pilot 开销的关系，基于模型给出了两元三次曲面方程，见方程 14，并给出了相应方程系数表，见表 3、表 4、表 6，同时还计算出不同定量情况下模型的曲面拟合优度，见表 5、表 7。通过拟合三维曲面图，判断三者关系为若 D_z 越小，或 LW 越小，则 Pilot 开销越小。

4.3 问题三分析

本题是在导频开销可任意变化的条件下，考虑芯片中定点量化对性能和资源的影响，目标是将芯片实现资源最低。由于在问题一中，不考虑定点量化时，设计的 CR 算法已经实现了题目中要求的性能 (RSNR 代价 $<0.3d$)，此时为了达到资源的最低，我们将关键点集中在：如何在达到题目性能的前提下，得到最小的位宽。

4.3.1 算法框架设计

由于题目要求最大并行度为 128，所以我们设计插入导频的方式为每 128 个信号为一组，前 127 个信号为有效信号，第 128 个信号为导频：



图 25 导频插入方法

输入的数据需要对在 CPU 中进行定点化处理，并将定点化后的数据通过数据总线流向各个模块。首先需要获得导频信号，并提取相位，得到导频信号的相位后，采用线性插值法获得每个有用数据待补偿的相位，最后进行相位补偿。下面给出数据整个处理流程，如图 26 所示。

整体框图设计分为软件部分与硬件部分。

软件部分仅实现色散补偿后的信号进行定点化处理，寻找满足性能的最小位宽，其中包括对色散补偿后的信号进行归一化处理，以及在保证性能的条件下，寻找最小位宽，以达到芯片资源占用最小的目标。

芯片部分需要实现以下四个子模块：

(1) 处理导频信号，用改进的 LUT 查表法，获得导频的相位 θ_a 。

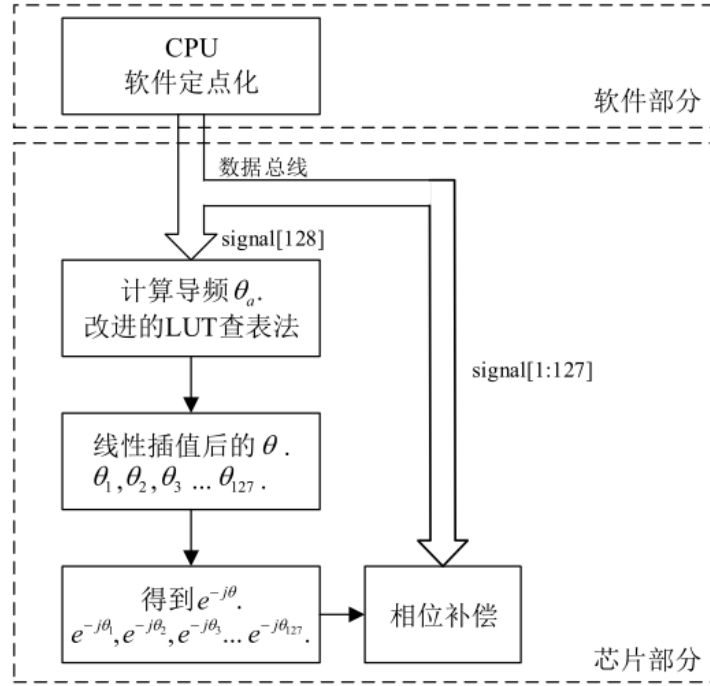


图 26 整体框图

- (2) 采用线性插值法，获得每个有效信号的相位。
- (3) 采用查表法，获得有效信号需要补偿的相位 $e^{-j\theta}$ 。
- (4) 对有效信号进行相位补偿。

接下来将对每个部分的设计思想进行详细描述，在对芯片部分的分析时，会给出 ASIC 芯片设计方案，并列出每个子模块相应的资源需求。

4.3.2 软件模块实现

软件模块需要实现定点化方案，在 CPU 中进行，其方案如下，首先对色散补偿后的信号进行归一化，接着对探索位宽与 BER 的关系，选择合适的位宽，对其进行定点化后，再送入到数据总线，实现相位补偿。

对色散补偿后的信号进行归一化的目的在于，可以使得所有信号的实部、虚部的范围在 $[-1,1]$ ，此时只需要用一位表示符号位，尽可能地减少了整数部分占的位宽。设任意两个导频及其有用信号如下：

$$A_a + iB_a, a_1 + ib_1, a_2 + ib_2, a_3 + ib_3, \dots, a_{127} + ib_{127}, A_b + iB_b \quad (16)$$

其中， $A_a + iB_a$ 表示上一组 127 个信号末尾插入的导频， $A_b + iB_b$ 表示本组 127 个信号末尾插入的导频。其中 a 和 b 表示有用信号，大写的 A 和 B 表示导频。对所有的信号归一化得：

$$\frac{a_1 + ib_1}{\sqrt{a_1^2 + b_1^2}}, \frac{a_2 + ib_2}{\sqrt{a_2^2 + b_2^2}}, \frac{a_3 + ib_3}{\sqrt{a_3^2 + b_3^2}}, \dots, \frac{a_{127} + ib_{127}}{\sqrt{a_{127}^2 + b_{127}^2}}, \frac{A_a + iB_a}{\sqrt{A_a^2 + B_a^2}} \quad (17)$$

此时得到归一化后的信号：

$$\alpha_1 + i\beta_1, \alpha_2 + i\beta_2, \alpha_3 + i\beta_3, \dots, \alpha_{127} + i\beta_{127}, \lambda_a + i\kappa_2 \quad (18)$$

其中， $\alpha_i, \beta_i, \lambda_a, \beta_a \in [-1, 1]$ 。需要符号位占 1 位，整数位占 1 位，小数位经过计算占 7 位。基于问题二场景，取 Dz 为 5000ps/nm，取 LZ 为 100kHz，此时仿真获得位宽与 BER 关系图如下图 27 所示。

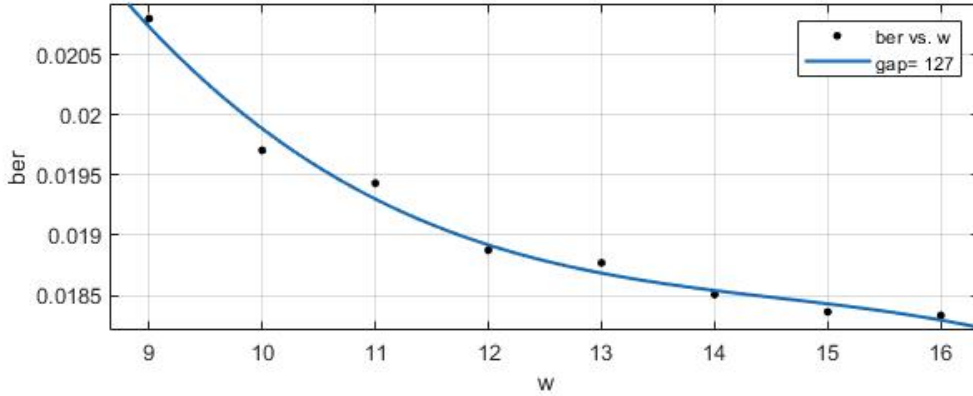


图 27 位宽与 BER 关系

这里需要注意，由于考虑到 θ 值的范围是 $[-\pi, \pi]$ ，所以需要多保留一位整数位位宽。

综上所述，我们采用的定点数位宽为 10 位，第 1 位表示符号位——0 表示正数，1 表示负数；第 2、3 位表示整数位；剩余 7 位表示为小数位。具体见图 28。

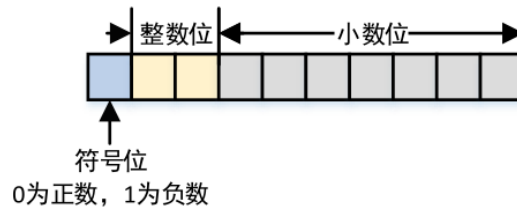


图 28 ASIC 中的定点位数表示

4.3.3 芯片模块实现

为了能够在 ASIC 芯片上计算导频相噪 θ ，我们使用改进后的 LUT 查表法。首先我们需要计算导频虚部比实部 $\frac{\kappa_a}{\lambda_a}$ 的值，接下来只需要根据反正切函数对应的表值进行查表即可得到需要的导频相噪。使用改进后的 LUT 查表法，相比于其它查表法的优势在于，首先 $\arctan \theta_a$ 是奇函数，其次 $\theta \in [0, \frac{\pi}{4})$ 和 $\theta \in [\frac{\pi}{4}, \frac{\pi}{2})$ 之间存在关系，只需要提前保存 $0 \leq \frac{\kappa_a}{\lambda_a} < 1$ 的反正切值，即存储 $\theta \in [0, \frac{\pi}{4})$ 的值，相比于一般的查表方法大大减少了 ROM 表的容量。此外由于 $\theta \in [0, \frac{\pi}{4})$ ，对应的正切值在 $[0, 1)$ ，只需要 9bit 位宽，大大减少了资源。

此时需要注意，由反正切函数求出的 $\theta_a \in (-\frac{\pi}{2}, \frac{\pi}{2})$ ，为了使在相位恢复时 $e^{j\theta}$ 能够取到理论上全部的值，需要保证 $\theta_a \in (-\pi, \pi)$ ，此时需要对改进后的 LUT 查表法进行改进。

对 LUT 查表法进行改进，对相噪 θ_a 进行求解，逻辑如下

$$X_a = \frac{\kappa_a}{\lambda_a} \Rightarrow \begin{cases} X_a > 0 \begin{cases} 0 \leq X_a < 1 \Rightarrow \text{查表得 } \theta_a \\ 1 \leq X_a < \infty, X_a = \frac{X_a-1}{X_a+1} \Rightarrow \text{查表得 } \theta_a \Rightarrow \text{修正为 } \theta_a + \frac{\pi}{4} \end{cases} \\ X_a < 0 \text{查表得 } X_a = X_a \times (-1) \begin{cases} 0 \leq X_a < 1, \text{查表得 } \theta_a \Rightarrow \text{修正为 } \theta_a \times (-1) \\ 1 < X_a \leq 1, \text{查表得 } \theta_a \Rightarrow \text{修正为 } \theta_a \times (-1) - \frac{\pi}{4} \end{cases} \end{cases} \quad (19)$$

查表后，对 θ_a 进行修正算法逻辑如下：

$$\theta_a = \frac{\kappa_a}{\lambda_a} \Rightarrow \begin{cases} X > 0 \begin{cases} \kappa_a > 0, \lambda_a > 0, \theta_a \text{不需要做修正} \\ \kappa_a < 0, \lambda_a < 0, \theta_a \text{修正为 } \theta_a - \pi \end{cases} \\ X < 0 \begin{cases} \kappa_a < 0, \lambda_a > 0, \theta_a \text{不需要做修正} \\ \kappa_a > 0, \lambda_a < 0, \theta_a \text{修正为 } \theta_a + \pi \end{cases} \end{cases} \quad (20)$$

改进后的 LUT 查表法，硬件逻辑如图 29、30 所示：

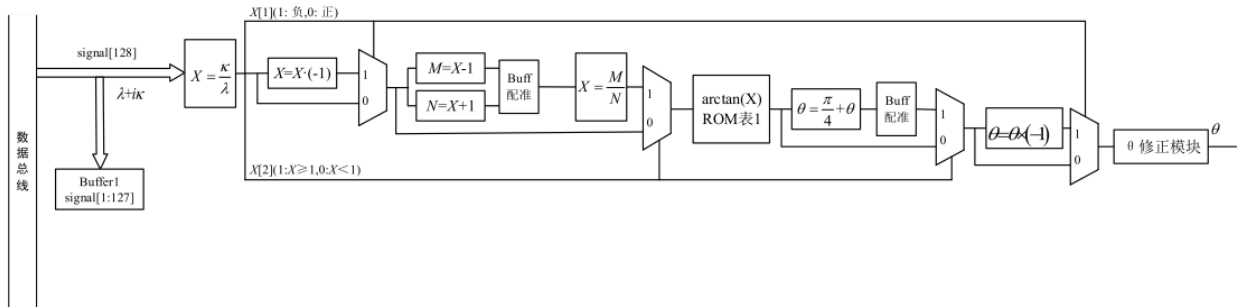


图 29 改进的 LUT 查表法硬件逻辑

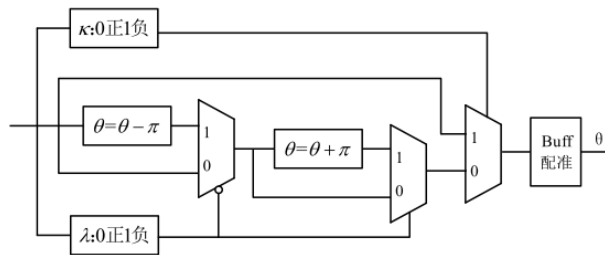


图 30 θ 修正模块内部逻辑

计算导频相位模块需要资源表 (该表只统计假设中的基本操作，该模块中设计的除法用查表实现，本表中的基本操作为实数加法器、实数乘法器、查表)：

表 8 导频相位模块资源表

基本操作	10+10bit	10*10bit	10bit-10bit	计算导频相位模块需要时间
个数	4	2	3	
资源	32U	128U	3072U	8 拍

在此举例说明该资源表的使用方法，基于本题，一个 8+8bit 复数加法器占用资源 1U，则 1 个 10+10bit 实数加法器计比 1 个 8+8bit 复数加法器多 4 位，则一个 10+10bit 复数加法器占用的资源是 $1 * 2^4 = 16U$ ，一个 10+10bit 实数加法器占用资源是 $\frac{1*2^4}{2} = 8U$ 。

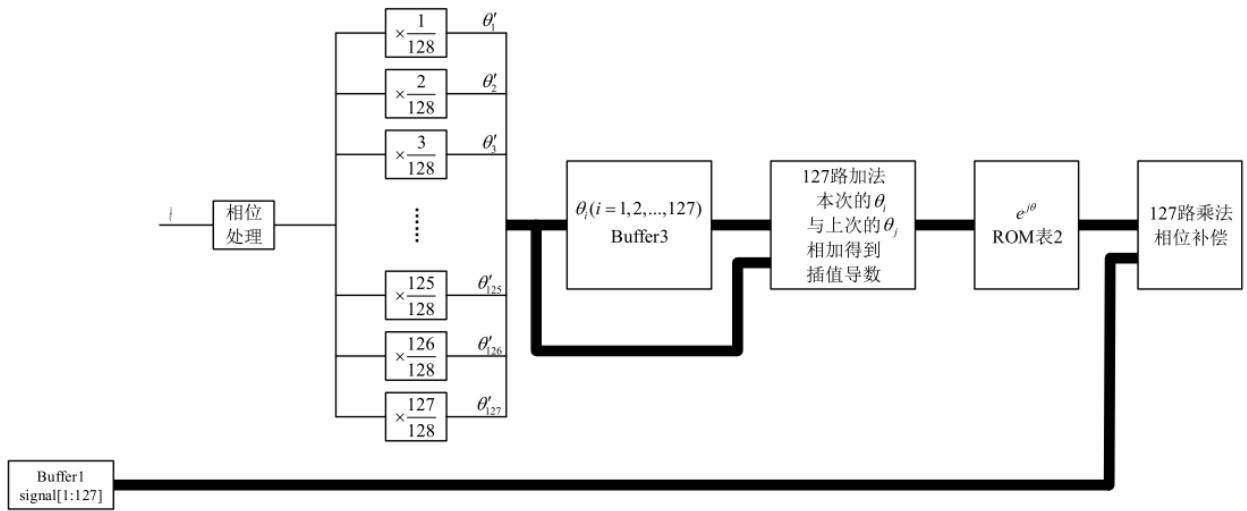


图 31 插值、查表、相位补偿模块硬件逻辑

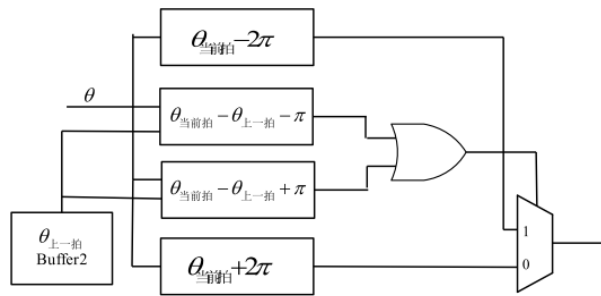


图 32 θ 相位处理模块内部逻辑

基于改进的 LUT 查表法完成导频相噪 θ_a 计算后，首先采用线性插值法，获得每个有效信号的相位，之后采用查表法，获得有效信号需要补偿的相位 $e^{-j\theta_i}$ ，最后对信号进行相位补偿。由于将所有的相位值限制在了 $(-\pi, \pi)$ 之间，那么相邻的两个相位 θ_1 和 θ_2 可能被分为 θ_1 与 $\theta_2 + 2k * \pi, k = \pm 1$ 。为了保证插值的连续，需要使连续的两个导频相位连续，不

能产生 2π 的跳变。将当前时刻的 θ 与上一时刻的 θ 作比较，若差值过大则通过增减 $2 * \pi$ 进行修正后才能插值。这三个部分的 ASIC 实现如图 31、图 32。

此时计算资源表见表 9，表中操作为复数加法器、复数乘法器、复数 ROM 表：

表 9 插值、查表、相位补偿模块资源表

基本操作	10+10bit	10*10bit	10bit-10bit	127*10bit 暂存 buffer	模块需要时间
个数	6	256	1	2	
资源	96U	32768U	2048U	0.62U	7 拍

在此举例说明该资源表的使用方法，由题目得，一个 8+8bit 复数加法器占用资源是 1U，则 6 个 10+10bit 复数加法器计算如下： $16 * 6 = 96U$ 。

4.3.4 优化分析

题目中要求尽可能减少所用资源，主要基于两方面考虑：位宽的优化和查表的优化。

A. 位宽优化

(1) 16QAM 的调制信号实部和虚部幅值最大为 3，在导频插入时，为了更好辨别导频的相噪，增大了导频的功率，整体信号在经过信道的色散和相噪以及白噪声叠加后，幅值可能增大到超过 16，即需要至少 5 个整数位进行存储。优化方案中在 CPU 处即对信号作了取模操作，将模值单独存放，数据进行归一化后进行处理。归一化后的数据无需占用整数位。

(2) 在求噪声相位时需要根据所得相噪虚部和实部的比值（相位正切值）来查表找到对应相位。正切值的取值范围为负无穷到正无穷），而 89 度角的切值也达到了 57，至少需要 6 个整数位。优化方案中通过正切的三角函数关系，将绝对值大于 1 的正切值转换到 (0,1) 内进行计算，转换后无需占用整数位。

(3) 相位噪声是累加的高斯白噪声，当线宽较大时其幅值也会相应等比例增大，幅值超过 8 则会占用至少 4 个整数位。由于 $e^{(i*\theta)} = e^{(i*(\theta+2k*\pi))}$ ，相位以 2π 为周期，则可以将其幅值限制在 $(-\pi, \pi)$ 内，只占用 2 个整数位。

综上，可以节约的整数位至少为 4 个。本方案中最终采取 10 比特位宽（1 符号位 +2 整数位 +7 小数位），节约了约 $4/14=28.57\%$ 的位宽资源。

B. 查表优化

在对正切值进行处理时，通过符号的提取和数学的运算简化了查表步骤，将表的大小变为原先的 1/4。

综上，可以得出资源占用最小的 CR 算法，具体代码见附录 A。

4.4 问题四分析

本题需要给出统筹性的“性能——资源”综合考虑下的算法设计思路，构造性能和资源的综合代价函数，给出一套自动优化位宽和实现性设计的方案。性能指标为 RSNR 为 0.3dB 的误码率，资源指标为位宽以及 Pilot 开销，位宽与硬件所消耗资源成正比，Pilot 开销与相邻 Pilot 间隔 gap 成反比。

在问题三的基础上，设定场景为 $Dz=5000\text{ps/nm}$ ， $LW=100\text{kHz}$ 。通过遍历 gap 数值获得 gap、位宽、BER 数值，根据仿真结果构造函数，见方程 15，此时方程系数见下表，拟合优度具体如下，SSE 为 $8.966\text{e-}08$ ，R-square 为 0.9953，Adjusted R-square 为 0.9915，RMSE 为 $9.028\text{e-}05$ 。此时拟合得到曲面图见图 33。

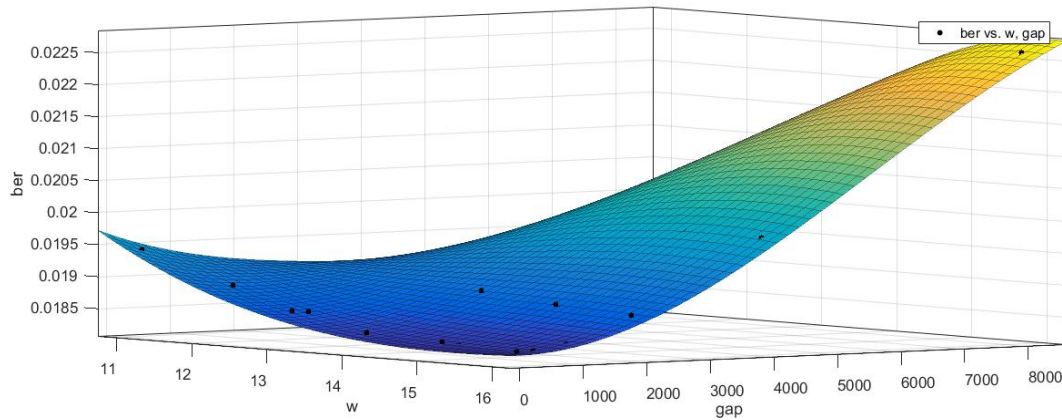


图 33 位宽、gap 与 BER 关系曲面图

表 10 位宽、gap 与 BER 关系曲面拟合系数

系数	数值
p_{00}	0.04564
p_{10}	-0.004685
p_{01}	-5.785e-06
p_{20}	0.0002626
p_{11}	7.881e-07
p_{02}	1.443e-11
p_{30}	-4.798e-06
p_{21}	-2.659e-08
p_{12}	6.952e-12
p_{03}	-7.682e-15

在众多仿真结果中，符合 BER 为 0.02 门限的数据组合见下表。

表 11 位宽、gap 与 BER 关系对应表

位宽	gap	BER
16	8191	0.0226
16	4095	0.0199
15	4095	0.0199
11	255	0.0194
14	2047	0.0190
12	511	0.0189
15	2047	0.0189
12	255	0.0188
16	2047	0.0188
13	255	0.0187
13	511	0.0186
13	1023	0.0185
14	1023	0.0185
14	255	0.0184
16	1023	0.0184
15	1023	0.0184
15	255	0.0184
14	511	0.0183
16	511	0.0183
15	511	0.0183
15	255	0.0183

根据表格数据，在相同的位宽下，较小的导频间隔之间差距不明显，较大的导频间隔会使得 BER 提升，系统性能下降，比如 gap 取 4000 至 8000。在相同的导频间隔下，位宽越大，性能越好。因此折衷采用较小的位宽和较大的导频间隔，因此基于表格挑选位宽为 13，对应导频间隔 gap 为 1023，此时 BER 为 0.0185，符合 0.02 门限。

这里需要注意，在相位补偿插值时采用乘法器，可精确到 $2^{-(\text{位宽}-3)}$ 位小数，因此导频间隔最大为 $2^{(\text{位宽}-3)} - 1$ 。

构造 BER 为位宽 w、gap 的函数 $\text{ber} = f(w, \text{gap})$ ，限制条件为 $\text{gap} \leq 2^{w-3} - 1$ 。此时构造

代价函数

$$c = \frac{w * ber}{\log_2(gap)} \quad (21)$$

故

$$w = \frac{c * \log_2(gap)}{ber} \quad (22)$$

在限定的误码率条件下，代价与导频间隔的 \log 值成反比。根据三次拟合， $w = f(c, \frac{\log_2(gap)}{ber})$ 中三次项和二次项的系数都非常小，可近似视为一次函数。拟合曲面图见图 34，拟合方程见方程 15，方程系数见表 12。拟合优度如下，SSE 为 0.000439，R-square 为 1，Adjusted R-square 为 1，RMSE 为 0.006273。

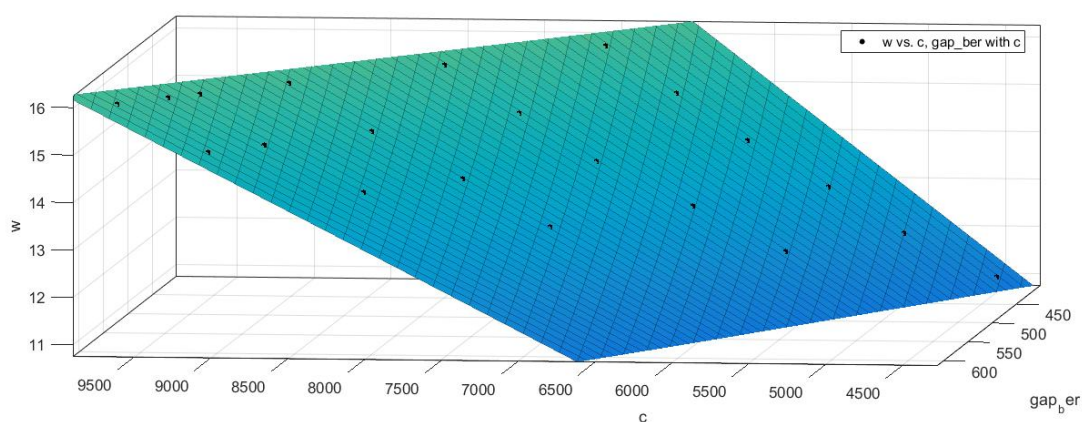


图 34 位宽、gap 与代价函数关系曲面图

表 12 位宽、gap 与代价函数关系曲面拟合系数

系数	数值
p_{00}	14.44
p_{10}	0.006079
p_{01}	-0.08801
p_{20}	1.477e-09
p_{11}	-1.228e-05
p_{02}	0.0001778
p_{30}	1.036e-12
p_{21}	-5.008e-11
p_{12}	8.95e-09
p_{03}	-1.228e-07

综上，系统可根据可承受代价的大小以及 gap 值来自动选取对应位宽。

参考文献

- [1] 刘洪微. 极低信噪比下基于连续相位调制的载波同步算法研究与实现 [D]. 西安电子科技大学, 2017
- [2] 沈丽丽. 数字通信系统中的载波恢复技术 [D]. 西安电子科技大学, 2004.
- [3] 王慧琴. 基于 FPGA 的 16QAM 调制模块的设计与实现 [J]. 自动化与仪器仪表, 2015(07):226-229.
- [4] 夏文娟. 基于高阶 QAM 的载波恢复方法的研究与实现 [D]. 合肥工业大学, 2013
- [5] 徐靖阳. NGB-W 信道估计二维维纳滤波算法研究及 FPGA 实现 [D]. 西安电子科技大学, 2018.
- [6] 杨国翔. QAM 系统中抑制相位噪声算法的研究 [D]. 西安电子科技大学, 2014.
- [7] 阳坚. 基于 FPGA 的 PM-QPSK 信号的实时相干解调算法的设计与实现 [D]. 华中科技大学, 2017
- [8] 张方正. 高速光通信中数字信号处理 (DSP) 与波形产生技术研究 [D]. 北京邮电大学, 2013.
- [9] 张肇敏. 高速相干光通信系统中高频谱效率调制及信号损伤恢复算法的研究 [D]. 北京邮电大学, 2015.
- [10] 周立丰. DVB-S 信道接收芯片载波恢复的研究及 ASIC 实现 [D]. 浙江大学, 2005.

附录 A 我的 MATLAB 源程序

1.1 性能最优 CR 算法设计

```
clear;clc;close all;
lameda = 1550*10.^-9;%波长
c = 3*1e8;%光速
fb = 150*10e9;%波特率
Dz = 20;%色散值
LW = 10e3;%线宽
SNR_db = sqrt(32.7)+0.3;
SNR = 10.^(SNR_db/10);
numBits = 1e7;%二进制数据长度
bits = round(rand(1,numBits));%二进制数据
data4 = reshape(bits, 4,numBits/4);%四个一组进行调制
data_sys = bi2de(data4.','left-msb');
data_mod = qammod(data_sys,16);%16qam调制
pilot = 12 * [1+1i,-1+1i,1-1i,-1-1i];
gap = 2000;%导频间隔
L_pilot = floor(numBits/4/gap)+1;%导频个数
pos = zeros(1,L_pilot);
pilot_pos = zeros(1,L_pilot);
for i = 1:L_pilot-1
    pos(i) = gap*i+i;%导频位置
    pilot_pos(i) = pilot(rem(i,4)+1);
end
pilot_pos(L_pilot) = pilot(rem(L_pilot,4)+1);%导频按顺序排列
pos(L_pilot) = numBits/4+L_pilot;%在最后补导频
data_pilot = zeros(1,length(data_mod)+L_pilot);
data_pilot(pos) = 1;
data_pilot(~data_pilot) = data_mod;%将导频插入数据
data_pilot(pos) = pilot_pos;
sigLen = length(data_pilot);
%%
data_fft = fft(data_pilot);%调制信号fft
Hf = exp(1j*(lameda.^2*pi*Dz.*(fb/sigLen:fb/sigLen:fb).^2/c));%色散fft
Dzadd_fft = data_fft.*Hf;%信道加色散
Xk = randn(1,sigLen);
dtheta = sqrt(2*pi*LW/fb)*Xk;
theta = zeros(1,sigLen);
```

```

mark = 0;
for i = 1:length(theta)
    if i == 1
        theta(i) = dtheta(i);
    else
        theta(i) = theta(i-1)+dtheta(i); %相噪
    end
end
theta_add = ifft(Dzadd_fft).*exp(1j*theta); %信号加相噪
noise_add = theta_add +
    sqrt(1/SNR)*(randn(1,sigLen)+1j*randn(1,sigLen)); %加高斯白噪声
Hf_conj = conj(Hf);
Dzdel_fft = fft(noise_add).*Hf_conj; %算法补色散
data_ifft = ifft(Dzdel_fft); %逆fft得到时域
theta_hat = zeros(1,sigLen);
for k = 1:L_pilot
    pilot_k = pilot(rem(k,4)+1);
    e_itheta = data_ifft(pos(k))/pilot_k;
    sig1 = sign(real(e_itheta)); %cos的正负号
    tan = imag(e_itheta)/real(e_itheta);
    sig2 = sign(tan); %tan的正负号
    if sig1 > 0
        theta_hat(pos(k)) = atan(tan);
    elseif sig2 > 0
        theta_hat(pos(k)) = atan(tan) - pi;
    else
        theta_hat(pos(k)) = atan(tan) + pi;
    end
end
if k == 1
    theta_hat(1:gap) =
        interp1([0,pos(k)],[0,theta_hat(pos(k))],1:gap); %插值
else
    delta = theta_hat(pos(k)) - theta_hat(pos(k-1));
    theta_hat1 = theta_hat(pos(k));
    while abs(theta_hat1-theta_hat(pos(k-1))) > pi %纠正不连续的theta
        if theta_hat1 > theta_hat(pos(k-1))
            theta_hat1 = theta_hat1 - 2*pi;
        else
            theta_hat1 = theta_hat1 + 2*pi;
        end
    end
end

```

```

        end
    end
    theta_hat((k-1)*(gap+1)+1:pos(k)-1) =
        interp1([pos(k-1),pos(k)],[theta_hat(pos(k-1)),theta_hat1],(k-1)*(gap+1)+1);
end
end
data_hat = data_ifft./exp(1j*theta_hat);
data_hat(pos) = [];%去除导频
data_demod = qamdemod(data_hat,16);%QAM解调
data_bit = de2bi(data_demod,'left-msb');
decBit = reshape(data_bit',numel(data_bit),1)';
ber = sum(bits ~= decBit)/numBits;%误码率计算

```

1.2 资源消耗最少 CR 算法设计

```

clear;clc;close all;
lameda = 1550*10.^-9;%波长
c = 3*1e8;%光速
fb = 150*10.^9;%波特率
Dz = 20;%色散值
LW = 1e5;%线宽
SNR_db = sqrt(32.7)+0.3;
SNR = 10.^(SNR_db/10);
numBits = 1e7;%二进制数据长度
bits = round(rand(1,numBits));%二进制数据
data4 = reshape(bits, 4,numBits/4);%四个一组进行调制
data_sys = bi2de(data4.','left-msb');
data_mod = qammod(data_sys,16);%16qam调制
pilot_f = 12 * [1+1i,-1+1i,1-1i,-1-1i];
num_f = 8:13;
for t = 1:6
    gap = 255;%固定导频间隔为255
    L_pilot = floor(numBits/4/gap)+1;%导频个数
    pos = zeros(1,L_pilot);
    pilot_pos = zeros(1,L_pilot);
    type = sfix(num_f(t)+3);%量化位数类型
    ff = 2.^-num_f(t); %量化到小数点后位数
    for i = 1:L_pilot-1
        pos(i) = gap*i+i;%导频位置
    end
end

```

```

        pilot_pos(i) = pilot_f(rem(i,4)+1);
    end
    pilot_pos(L_pilot) = pilot_f(rem(L_pilot,4)+1); %导频按顺序排列
    pos(L_pilot) = numBits/4+L_pilot; %在最后补导频
    data_pilot = zeros(1,length(data_mod)+L_pilot);
    data_pilot(pos) = 1;
    data_pilot(~data_pilot) = data_mod; %将导频插入数据
    data_pilot(pos) = pilot_pos;
    sigLen = length(data_pilot);
    %%
    data_fft = fft(data_pilot); %调制信号fft
    Hf =
        exp(1j*(lameda.^2*pi*Dz.*(fb/sigLen:fb/sigLen:fb).^2/c)); %色散fft
    Dzadd_fft = data_fft.*Hf; %信道加色散
    Xk = randn(1,sigLen);
    dtheta = sqrt(2*pi*LW/fb)*Xk;
    theta = zeros(1,sigLen);
    mark = 0;
    for i = 1:length(theta)
        if i == 1
            theta(i) = dtheta(i);
        else
            theta(i) = theta(i-1)+dtheta(i); %相噪
        end
    end
    theta_add = ifft(Dzadd_fft).*exp(1j*theta); %信号加相噪
    noise_add = theta_add +
        sqrt(1/SNR)*(randn(1,sigLen)+1j*randn(1,sigLen)); %加高斯白噪声
%    noise_add_fit =
    num2fixpt(real(noise_add),sfix(num_f(t)+5),2.^-num_f(t),'nearest')+1i*num2
    Hf_conj = conj(Hf);
    Dzdel_fft = fft(noise_add).*Hf_conj; %算法补色散
    data_ifft = ifft(Dzdel_fft); %逆fft得到时域
    abs_data_ifft = abs(data_ifft);
    data_unit = data_ifft./abs_data_ifft; %信号归一化
    abs_data_ifft(pos) = [];
    data_f = 1i*num2fixpt(imag(data_unit),type,ff,'nearest') +
        num2fixpt(real(data_unit),type,ff,'nearest'); %数据定点
    pilot =

```

```

    num2fixpt(pilot_f./abs(pilot_f),type,ff,'nearest');%pilot归一化后定点
theta_hat = zeros(1,sigLen);
for k = 1:L_pilot
    pilot_k = pilot(rem(k,4)+1);
    e_itheta = data_f(pos(k))/pilot_k;
    sig1 = sign(real(e_itheta));%cos的正负号
    tan = imag(e_itheta)/real(e_itheta);
    sig2 = sign(tan);%tan的正负号
    if abs(tan)>1
        tan_d = (abs(tan)-1)/(abs(tan)+1);%将tan值固定在-1到1
        tan_f = num2fixpt(tan_d,type,ff,'nearest');%tan定点
        theta_45 =
            sig2*(atan(tan_f)+0.25*pi);%恢复theta, theta从-0.5pi到0.5pi
        if sig1 > 0
            theta_hat(pos(k)) = theta_45;
        elseif sig2 > 0
            theta_hat(pos(k)) = theta_45 - pi;
        else
            theta_hat(pos(k)) = theta_45 + pi;%theta恢复到-pi到pi
        end
    else
        tan_f = num2fixpt(tan,type,ff,'nearest');
        if sig1 > 0
            theta_hat(pos(k)) = atan(tan_f);
        elseif sig2 > 0
            theta_hat(pos(k)) = atan(tan_f) - pi;
        else
            theta_hat(pos(k)) = atan(tan_f) + pi;
        end
    end
    theta_hat(pos(k)) =
        num2fixpt(theta_hat(pos(k)),type,ff,'nearest');%导频theta定点
    if k == 1
        theta_hat(1:gap) =
            interp1([0,pos(k)],[0,theta_hat(pos(k))],1:gap);%插值
        theta_hat(1:gap) =
            num2fixpt(theta_hat(1:gap),type,ff,'nearest');
    else
        delta = theta_hat(pos(k)) - theta_hat(pos(k-1));
    end
end

```

```

theta_hat1 = theta_hat(pos(k));
while abs(theta_hat1-theta_hat(pos(k-1))) >
    pi%纠正不连续的theta
    if theta_hat1 > theta_hat(pos(k-1))
        theta_hat1 = theta_hat1 - 2*pi;
    else
        theta_hat1 = theta_hat1 + 2*pi;
    end
end
theta_hat((k-1)*(gap+1)+1:pos(k)-1) =
    interp1([pos(k-1),pos(k)], [theta_hat(pos(k-1)),theta_hat1], (k-1)*(gap+1)+1:pos(k)-1);
theta_hat((k-1)*(gap+1)+1:pos(k)-1) =
    num2fixpt(theta_hat((k-1)*(gap+1)+1:pos(k)-1), type, ff, 'nearest');%数
end
end
data_hat_unit = data_f./exp(1j*theta_hat);%恢复归一化数据
data_hat_unit(pos) = [];%去除导频
data_hat_unit_f =
    num2fixpt(data_hat_unit, type, ff, 'nearest');%归一化数据定点
data_hat = data_hat_unit_f.*abs_data_ifft;%用绝对值恢复原始数据
data_demod = qamdemod(data_hat, 16);%QAM解调
data_bit = de2bi(data_demod, 'left-msb');
decBit = reshape(data_bit', numel(data_bit), 1)';
ber(t) = sum(bits ~= decBit)/numBits;%误码率计算
% end
end

A = theta_hat(pos); B = theta(pos);

```